(12) UK Patent Application (19) GB (11) 2 155 671 A

(43) Application published 25 Sep 1985

(21) Application No 8510106

(22) Date of filing 2 Feb 1983

Date lodged 19 Apr 1985

(30) Priority data
(31) 57/025616 (32) 19 Feb 1982 (33) JP

(60) Derived from Application No 8302850 under Section 15(4) of the Patents Act 1977

(51) INT CL⁴
G06F 13/00

(52) Domestic classification
G4A CX
U1S 1948 2097 G4A

(56) Documents cited
- None

(58) Field of search
G4A

(71) Applicant
Sony Corporation (Japan),
7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo, Japan

(72) Inventor
Osamu Hamada

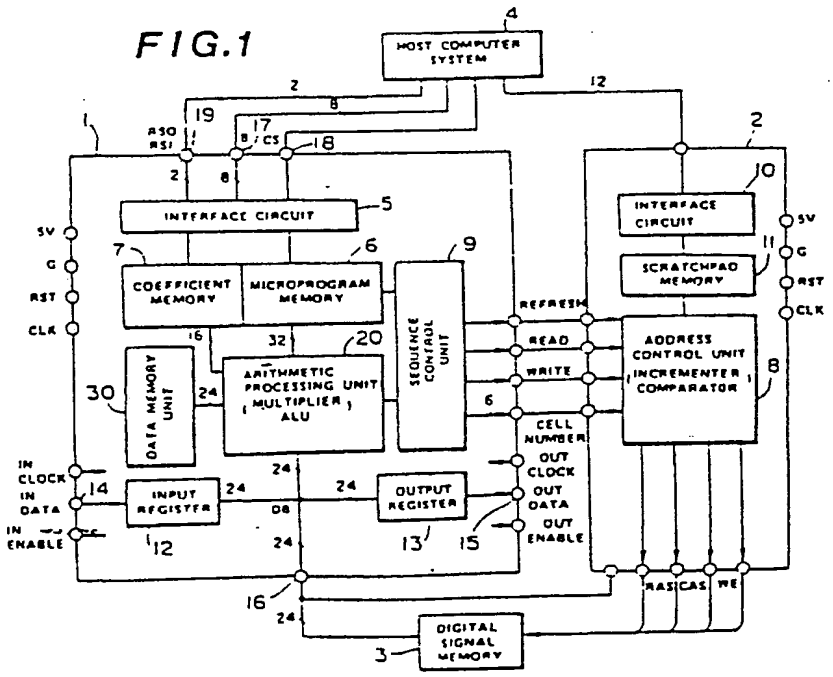
(74) Agent and/or Address for Service
D Young & Co,
10 Staple Inn, London WC1V 7RD

Reprinted front page

(54) Digital signal processing systems

(57) A digital signal processor (1) includes a microprogram memory (6) for storing a series of microinstructions for instructing a digital signal processing procedure, and a coefficient memory (7) for storing coefficient data required for performing a series of arithmetical operations on the digital signal data. Data is transferred and written into the microprogram memory (6) and coefficient memory (7) from a host computer system (4). The coefficient memory (7) has at least two pages corresponding to the total memory area which may be addressed during digital signal processing to be effected by the microinstructions. Page selection of the coefficient memory (7) may be performed from the host computer system (4).

FIG.1



2155671

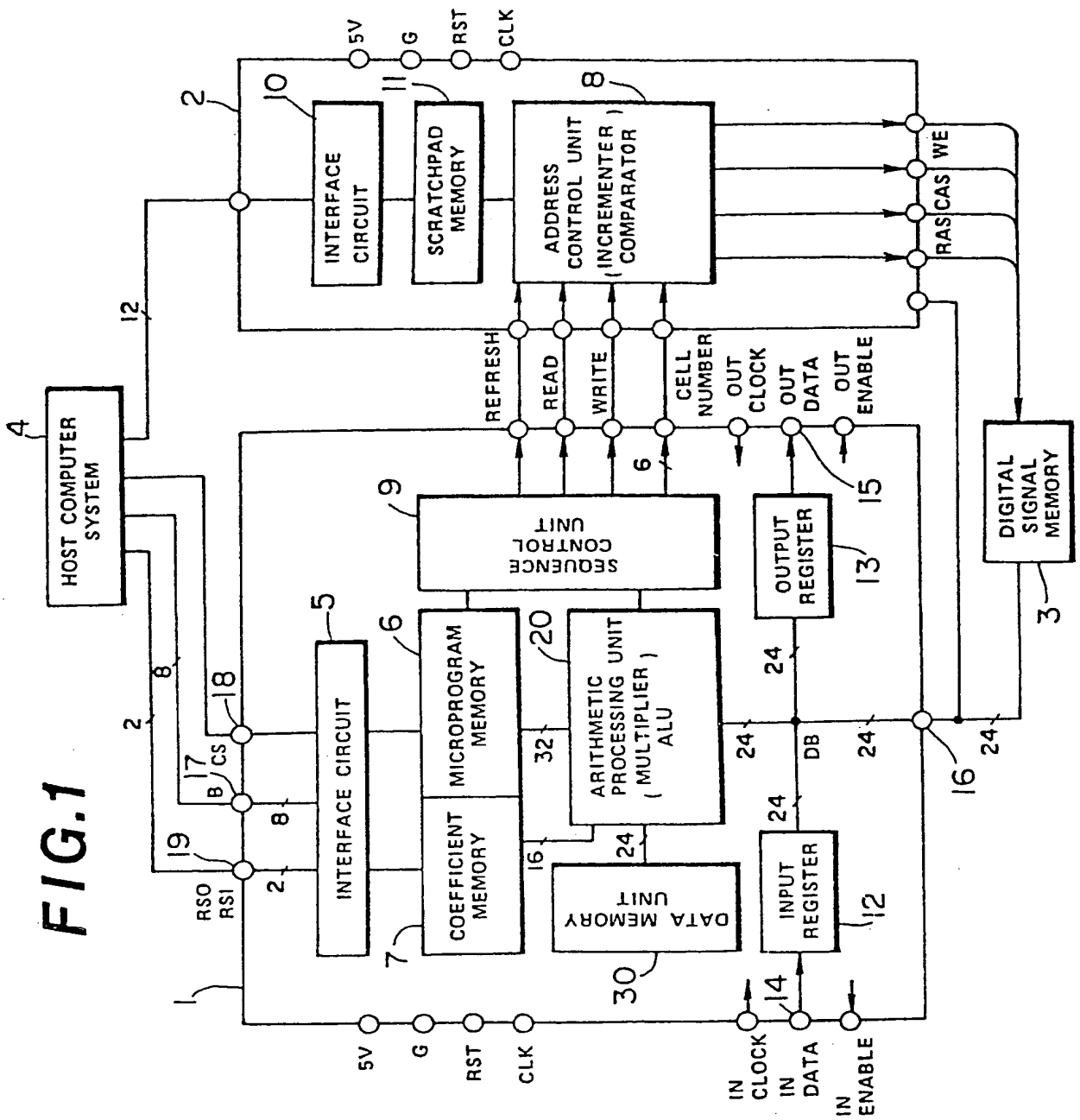


FIG.1

2155671

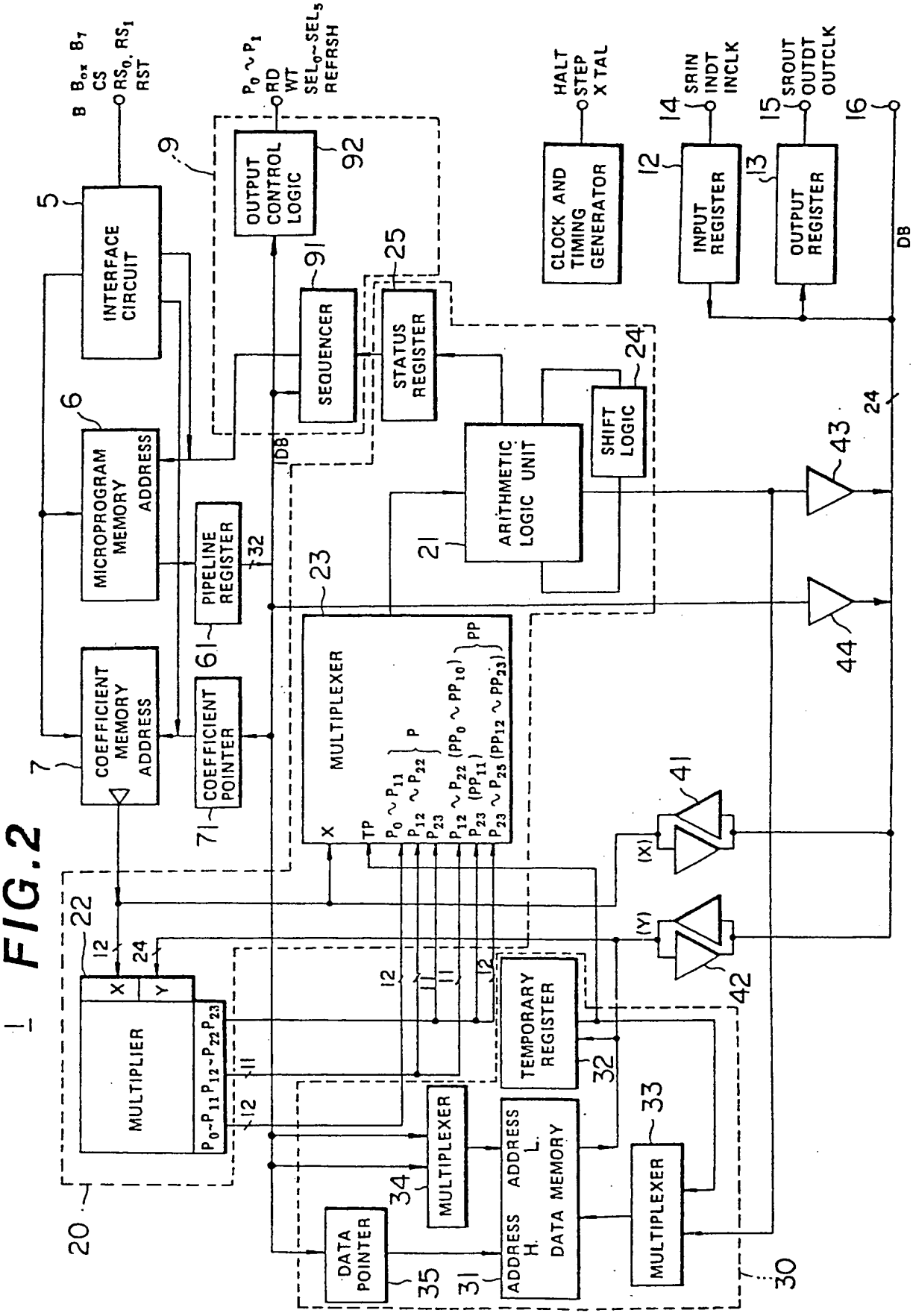


FIG.2

2155671

FIG. 3

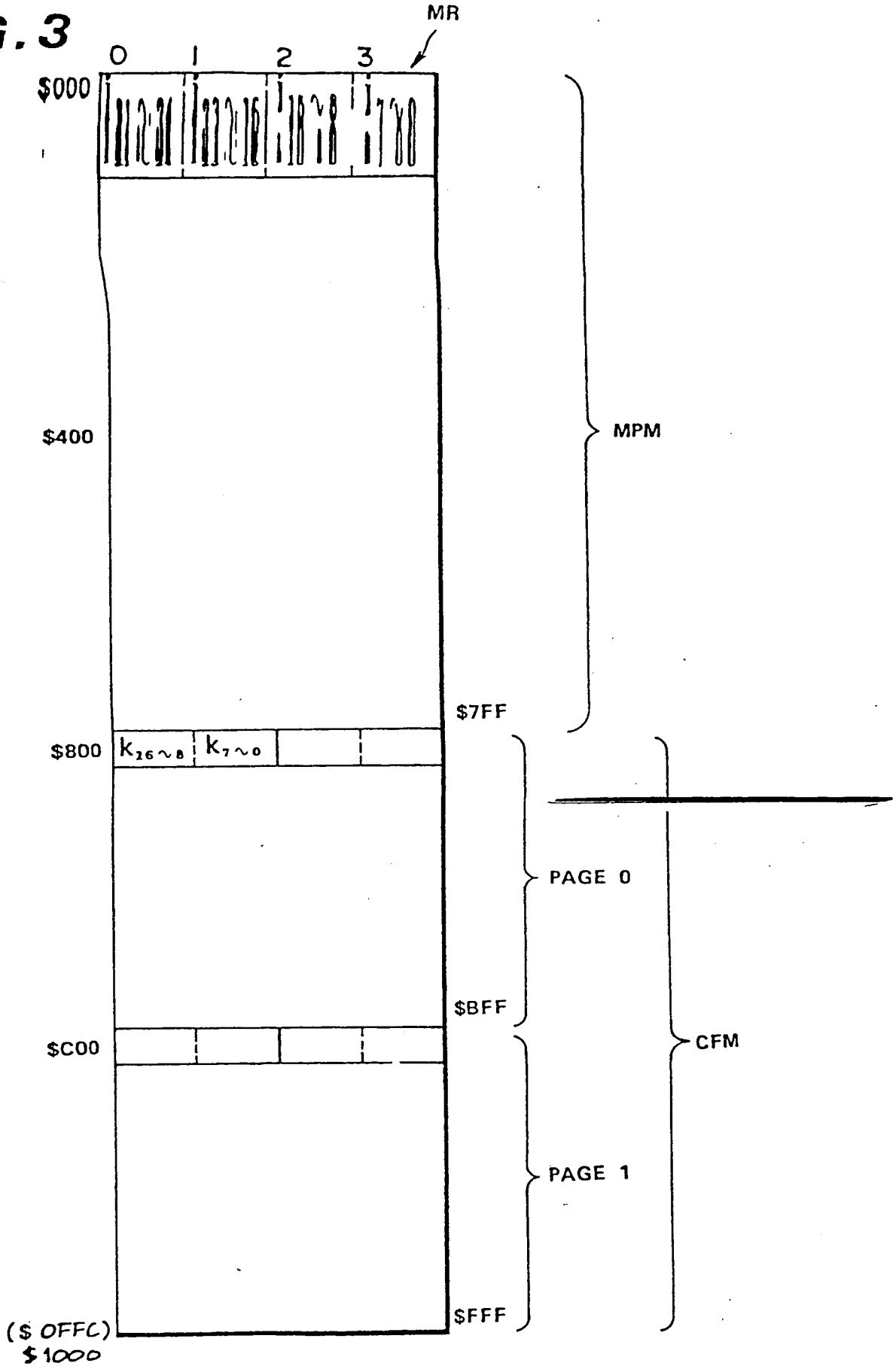


FIG.4

MODE		RSI	RSO	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
M_0	DATA	0	0	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
M_1	UPPER ADDRESS	0	1	X	X	X	X	a_{11}	a_{10}	a_9	a_8
M_2	LOWER ADDRESS	1	0	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
M_3	CONTROL	1	1	PAGE	$\overline{RST}$	$\overline{HALT}$	$\overline{STEP}$			FI	FO

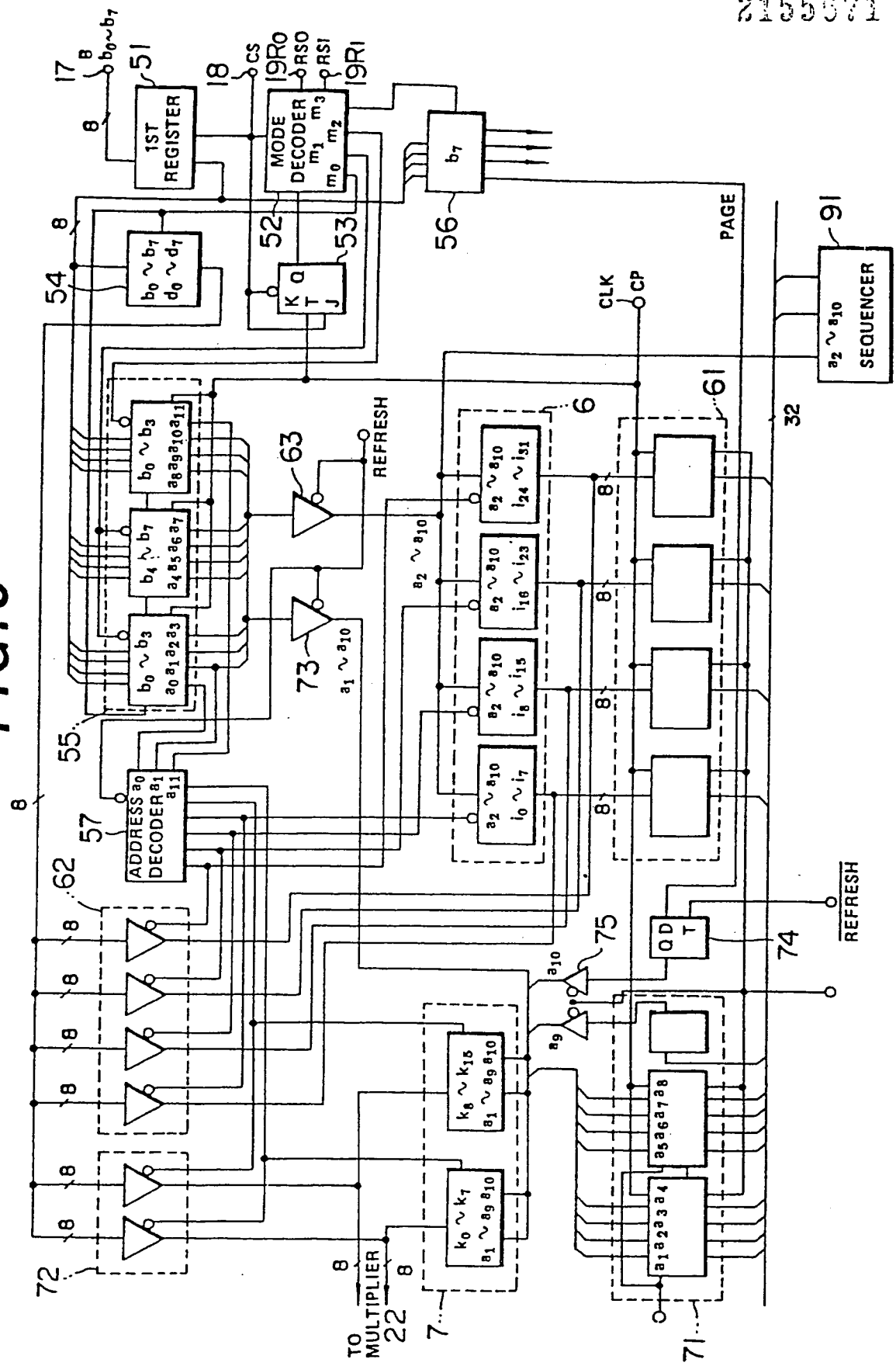
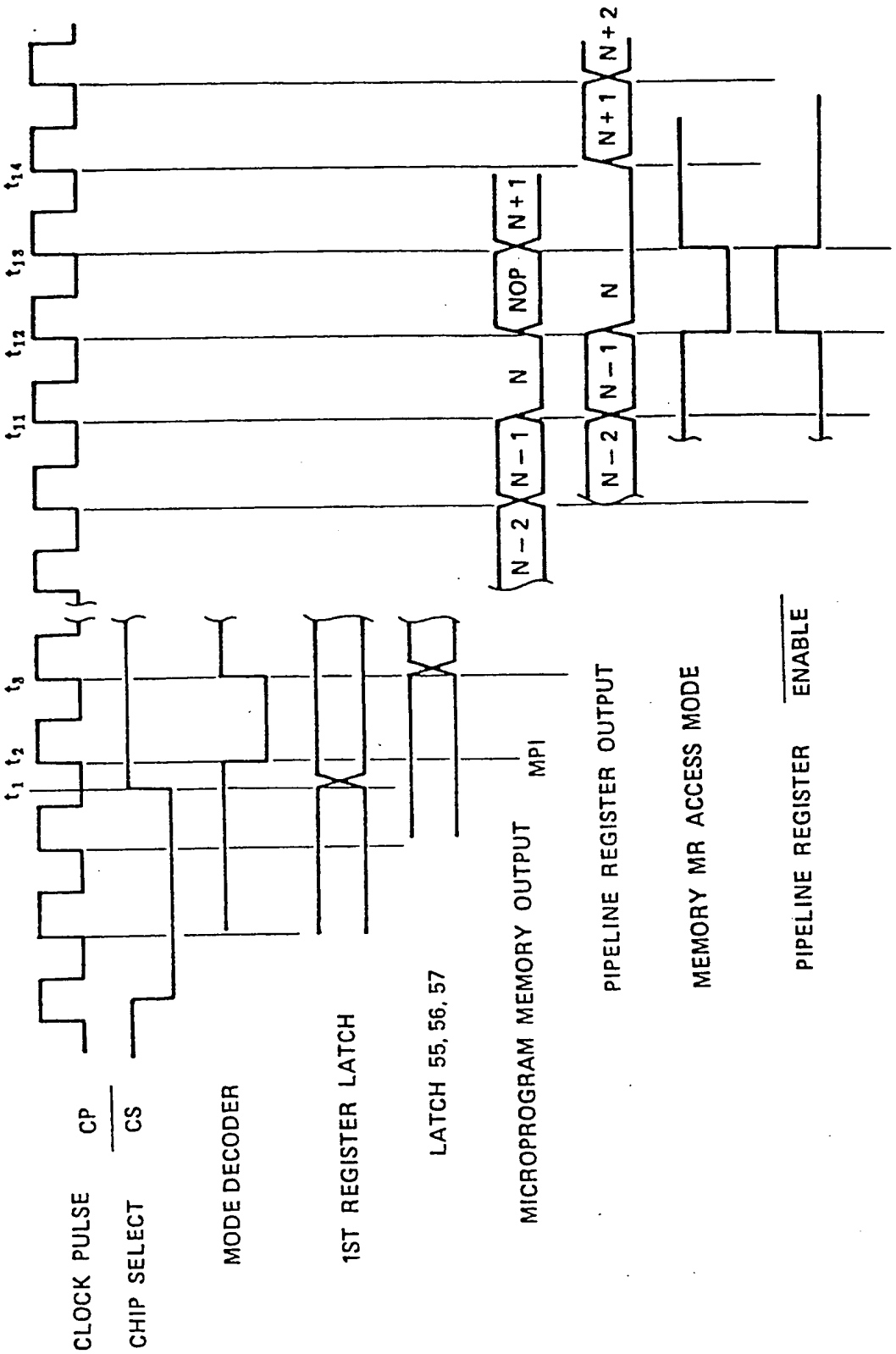


FIG. 5

FIG.6



2155671

FIG.7

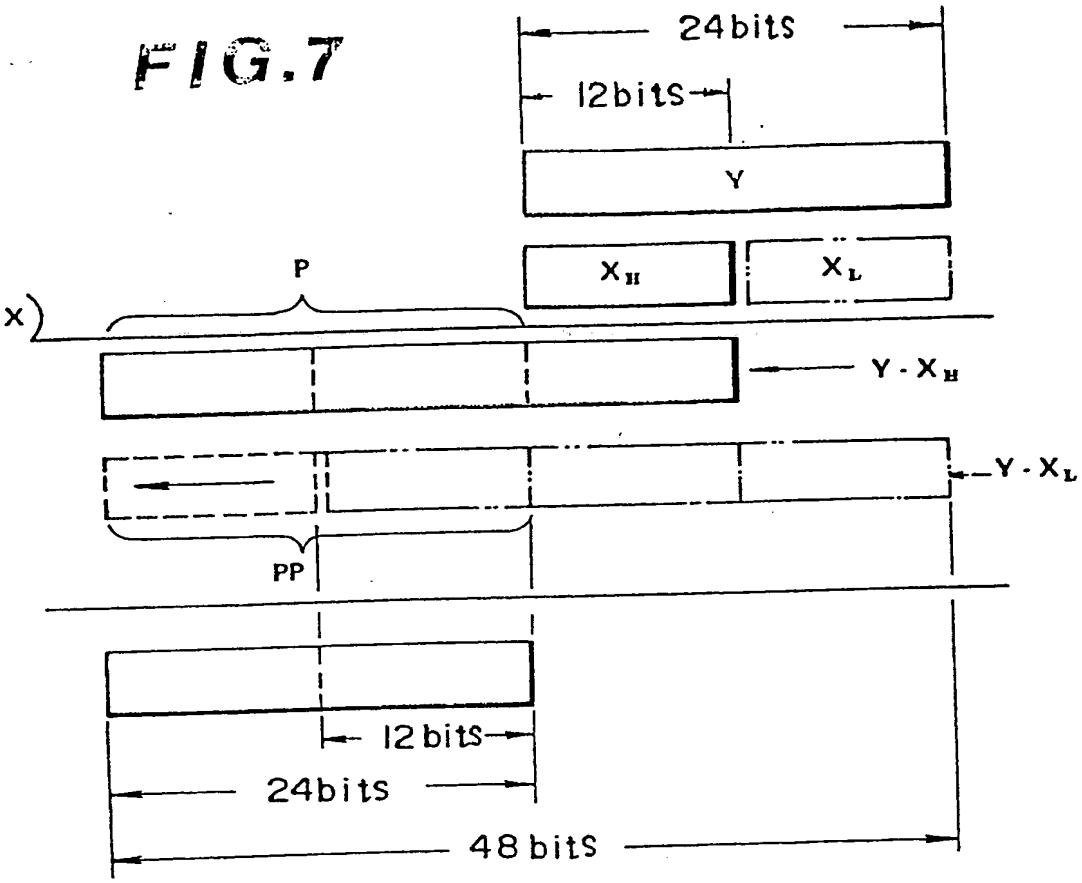


FIG.9

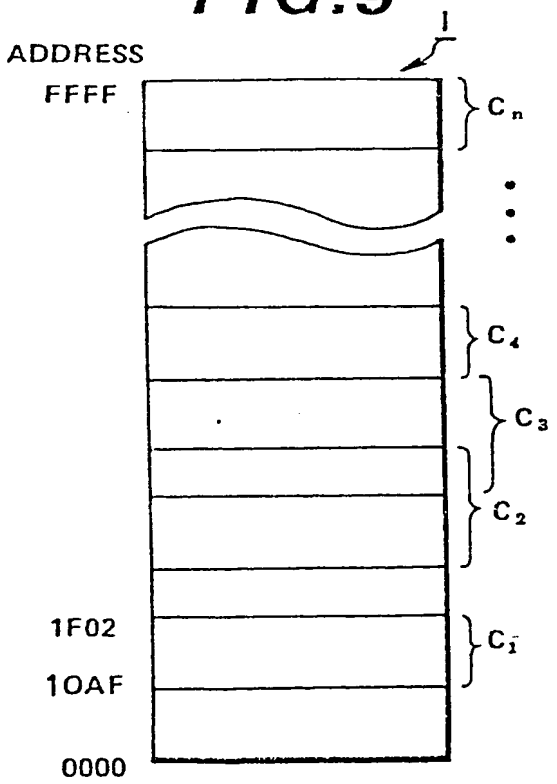


FIG.8

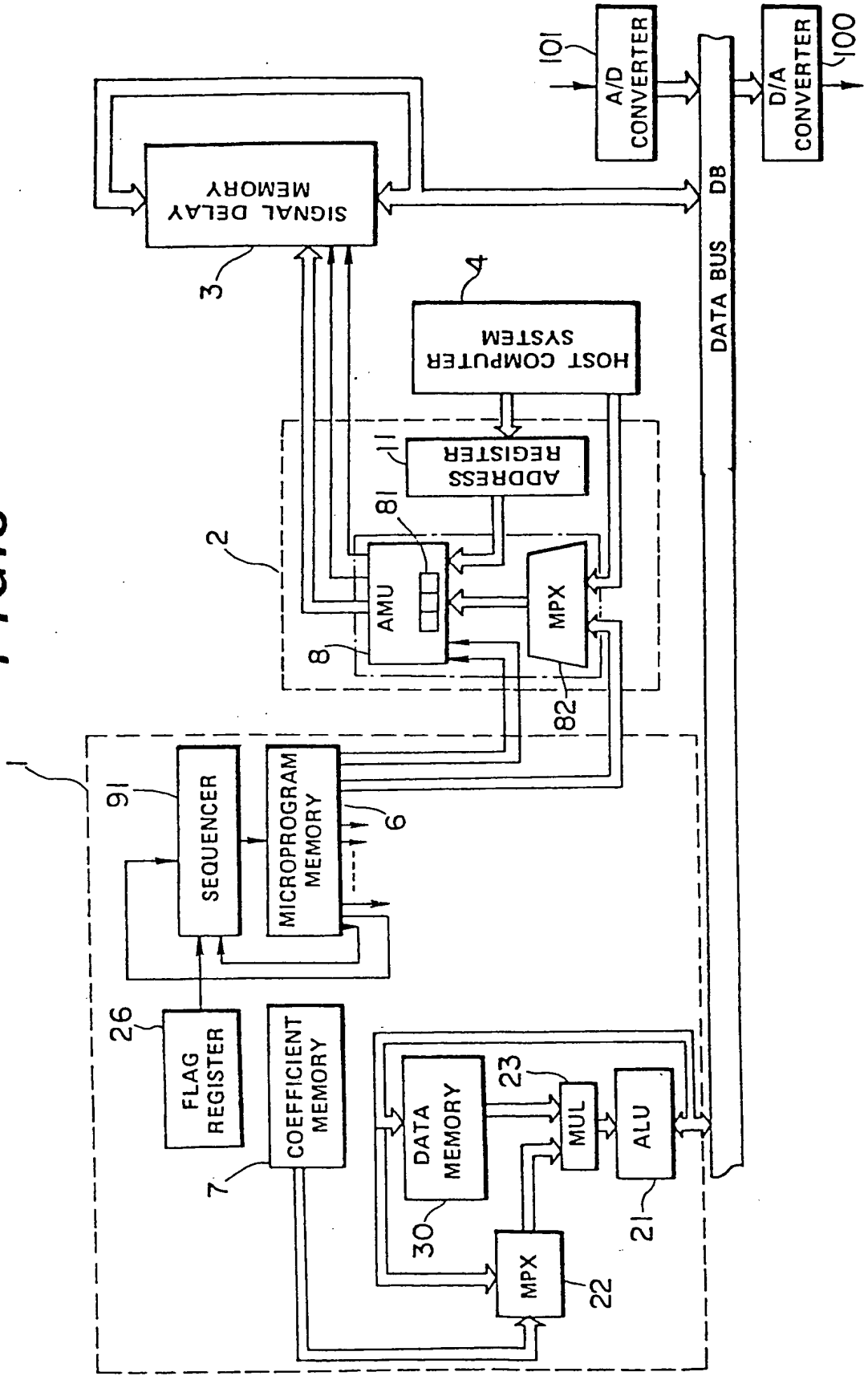
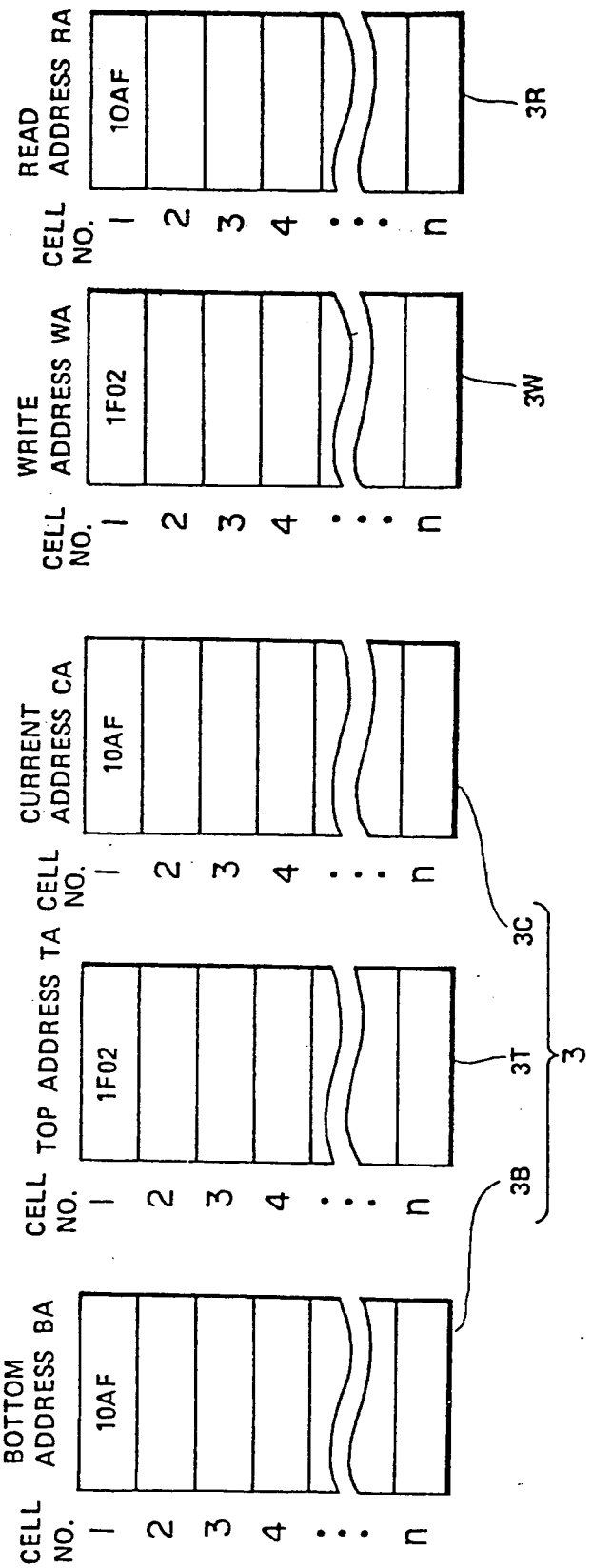
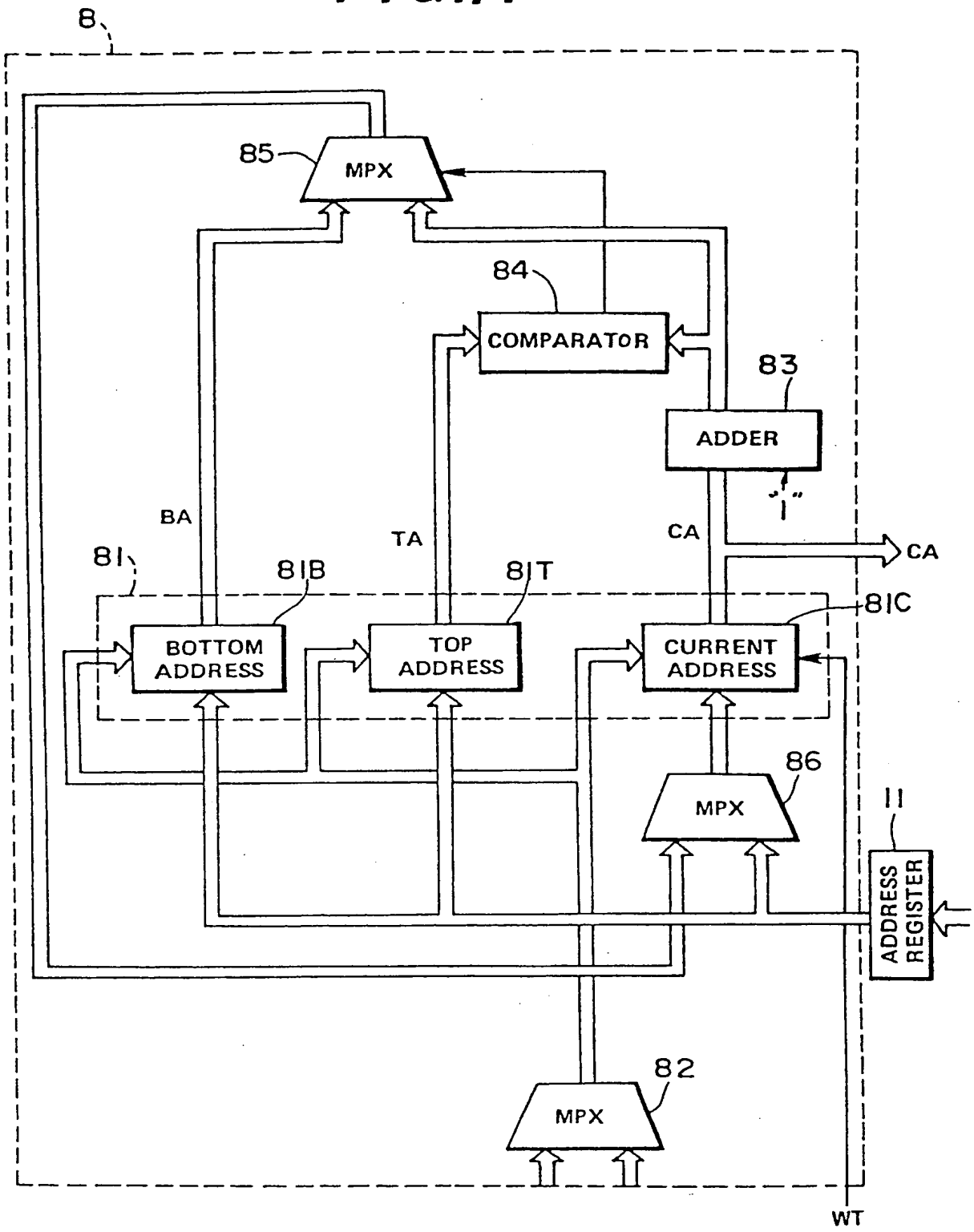


FIG.10



2155671

FIG.11



2155671

FIG.12

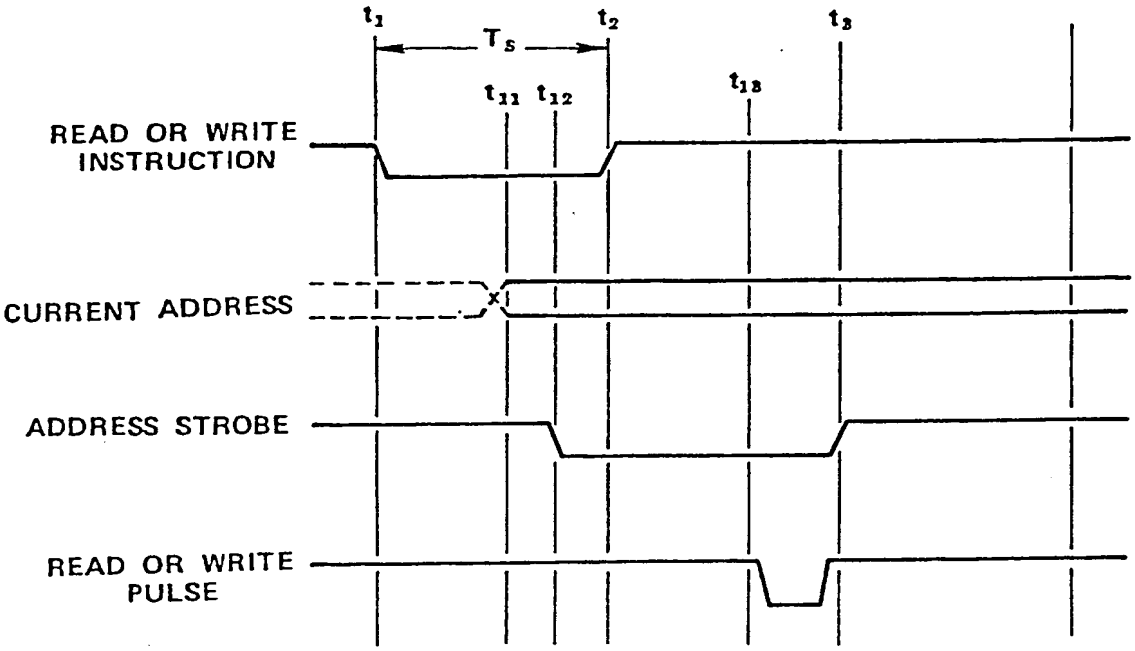


FIG.16

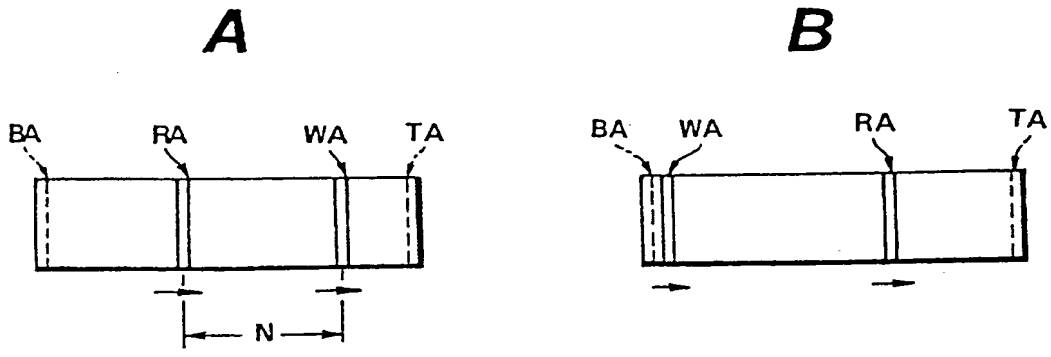


FIG.13

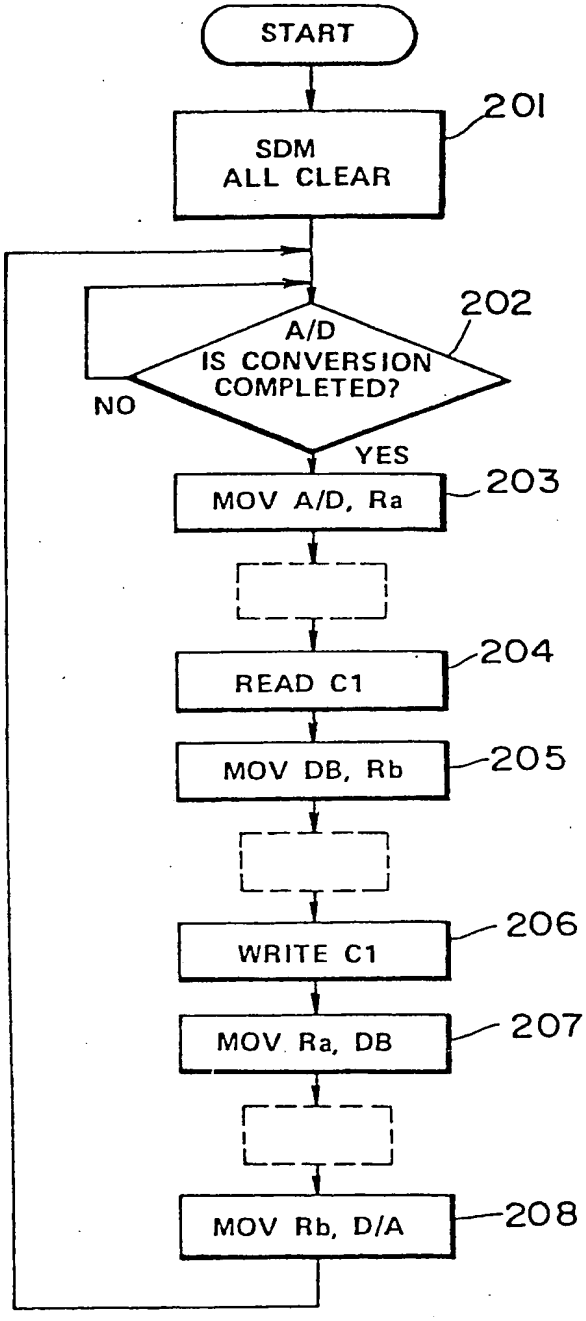


FIG.14

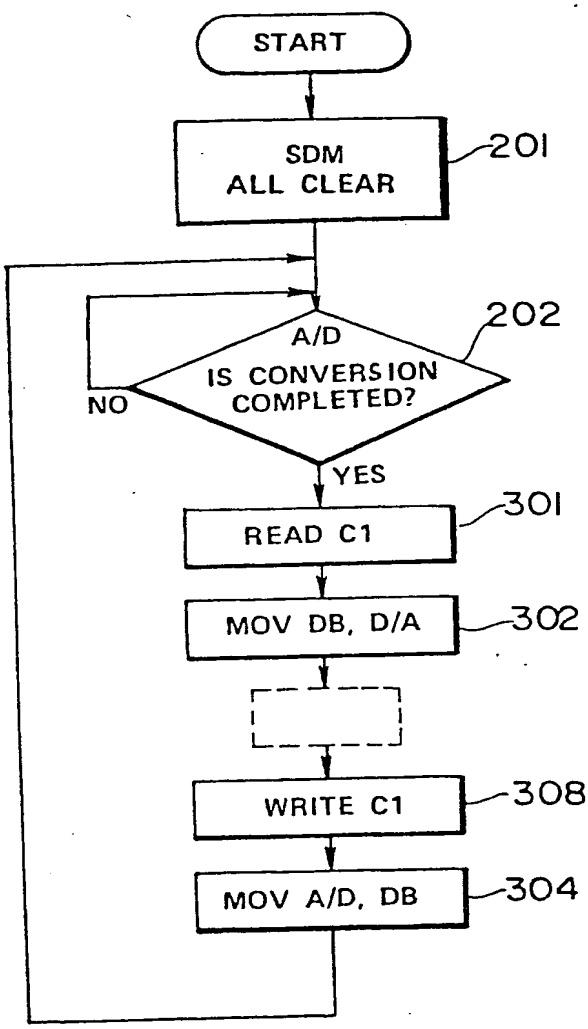
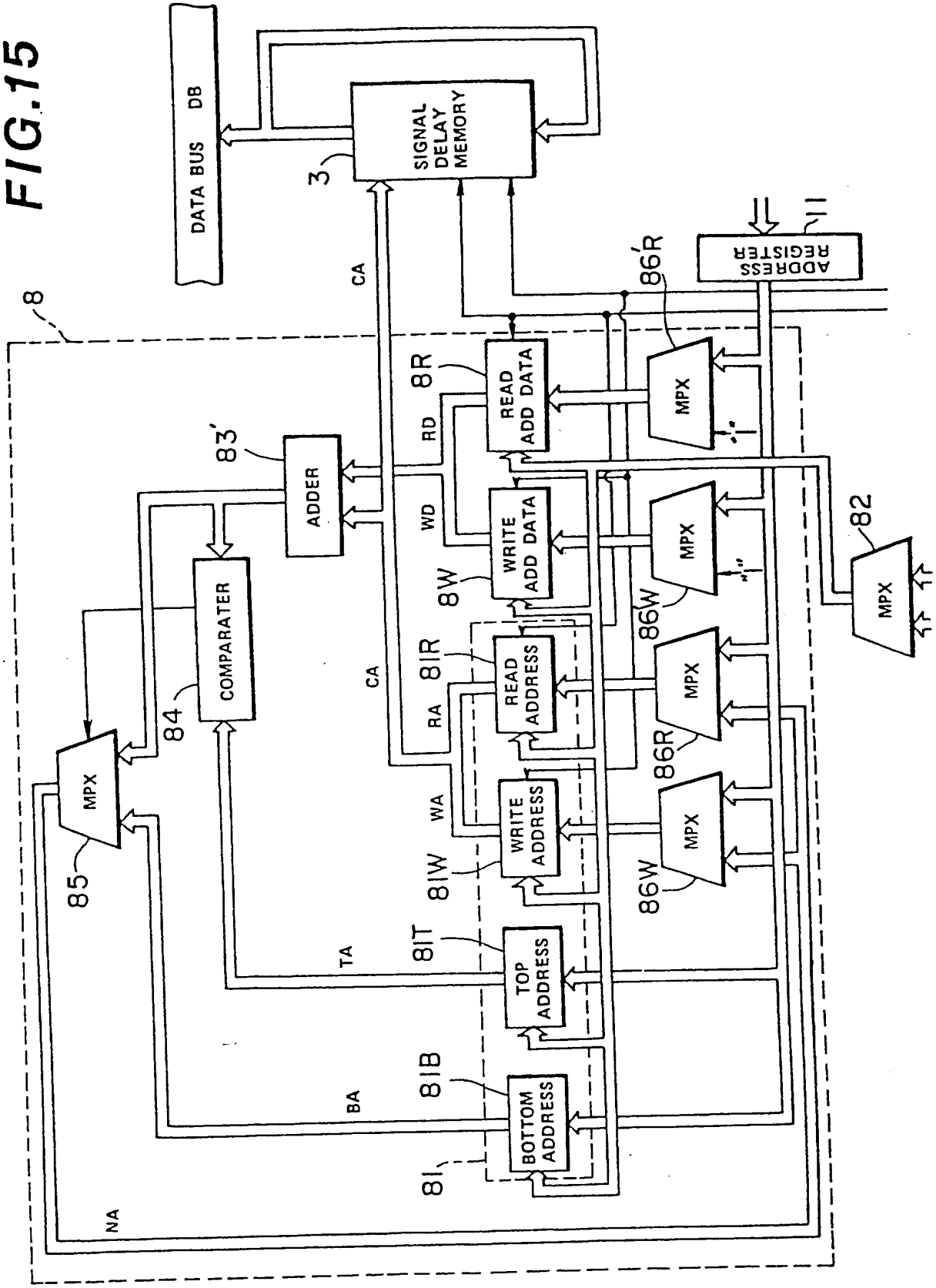


FIG.15



SPECIFICATION

Digital signal processing systems

This invention relates to digital signal processing systems. More particularly, but not exclusively, the invention relates to digital signal processing systems capable of real time processing of digital signals having a large number of bits per word and a high word rate per unit time, such as pulse code modulated (PCM) audio signals, and applicable for example to graphic equalisers or echo chambers having a digital signal delay feature.

In recent years, audio or picture signals have been converted into digital signals which then undergo a variety of digital processing or computing operations such as processing by digital filters, fast Fourier transformation (FFT) or correlative function computing on a real time basis. A variety of digital signal processors, hereafter sometimes abbreviated to "DSP", have been proposed for performing such real time processing operations. The DSPs are usually provided with high precision arithmetic logic units (ALUs), multipliers or other hardware units and controlled by microprograms. In many cases, such digital signal processing is controlled or managed by a host computer system making use of a microprocessor.

The DSPs are provided with internal memories, namely a microprogram memory and a coefficient memory. The signal processing operation in the DSPs is usually performed in such a manner that microinstructions stored in the microprogram memory are read out sequentially by addresses designated by sequencers or program counters.

System versatility is increased by using random access memories (RAMs) for the microprogram memory and coefficient memory, and the data to be entered into these memories may be transferred under the control of the host computer system.

In general, real time processing of digital signals converted from analog audio or picture signals necessitates fast arithmetical operations such as addition and multiplication, signal delaying and like processing operations. It has been customary to use multistage shift registers as hardware or circuitry for causing a delay in the digital signals. In this case, the delay time is expressed as a product of the number of the shift register stages and the sampling period or interval, that is the period of shift clock pulses.

However, a digital signal delay circuit designed for real time processing and making use of such shift registers is not well suited to changing of the delay time as desired during real time processing because of the necessity to change the number of shift registers. In addition, echo chambers or machines formed by a plurality of signal delay lines or circuits tend to have a complicated hardware structure. Above all, with delay circuits making use of the shift registers, it is in practice impossible to effect a dynamic change of the delay time in the respective delay lines. In addition, the use of a plurality of shift registers is not economical.

In the conventional DSPs, rewriting the coefficient data, microinstructions etc. during microprogram execution is a desideratum in order to increase the efficiency in real time processing. However, this is not in practice feasible because of oscillations caused by the data becoming discrete, especially in the course of rewriting the coefficient data.

In addition, when the digital signal data is multiplied in the DSP by coefficient data supplied from the coefficient memory, insufficient word length of the coefficient data may detract from the desired precision of the product, especially as a function of the characteristics of digital filters provided in the DSP. On the other hand, multiplier circuits may be extremely complex when the bit numbers of the multiplier and multiplicand are increased to 20 or larger.

According to the present invention there is provided a digital signal processing system comprising a digital signal processing unit including at least a microprogram memory for storing a series of microinstructions for instructing a digital signal processing procedure, and a coefficient memory for storing coefficient data required for performing a series of arithmetical operation on the digital signal data, a memory control unit, at least one memory block for storage of digital signals, and a host computer system, wherein

data may be transferred and written into the microprogram memory and the coefficient memory from the host computer system,

the coefficient memory has at least two pages corresponding to the total memory area to which access may be had during digital signal processing to be effected in response to the microinstructions,

page selection of the coefficient memory may be performed under control from the host computer system,

the memory control unit has an address management memory for dividing said one memory block into a plurality of memory cells identified by respective addresses,

means are provided for writing boundary addresses of the memory cells from the host computer system into the address management memory in advance, and

memory access is made with serial memory cell numbers designated by the digital signal processing unit.

Preferred systems embodying the invention and described in detail below possess the following advantageous features. Plural signal delay circuits can be realised with a simple hardware design. The number and delay time of the delay circuits can be set through programming as desired. The data stored in the coefficient memory or microprogram memory can be changed from the host computer system during execution of the microprogram. Discrete coefficient data is not produced. Oscillations or other undesirable effects may be prevented from occurring. The coefficient data may be updated during digital signal processing. The preferred systems may have improved efficiency in multiplication of the digital signal data by the coefficient data. The preferred systems may be provided with a smaller size multiplier and relatively fast high precision multiplication may be performed without unduly prolonging the processing time, even in cases where multiplication has be to performed with a long word length or high precision coefficient data.

The invention will now be further described, by ay of illustrative and non-limiting example, with reference to the accompanying drawings, in which the same references indicate corresponding items throughout, and in which:

Figure 1 is a block diagram showing a basic system configuration making use of a digital signal processor (DSP);

Figure 2 is a block diagram diagrammatically showing the internal structure of the DSP;

Figure 3 shows a memory map of a microprogram memory and a coefficient memory of the DSP;

Figure 4 shows data transfer modes from a host computer system and the information content of data bits;

Figure 5 is a block diagram of an interface circuit, microprogram memory and coefficient memory with neighbouring circuit portions of the DSP;

Figure 6 is a time chart for illustrating the operation of the circuit of Fig. 5;

Figure 7 is a chart illustrating a multiplying operation performing in a multiplier in the DSP;

Figure 8 is a block diagram showing an example of an echo chamber or machine constructed from a plurality of digital signal delay circuits;

Figure 9 shows a memory map of a signal delay memory (SDM);

Figure 10 is a chart illustrating the structure of an address management memory;

Figure 11 is a block circuit diagram showing the internal structure of the address management memory;

Figure 12 is a time chart illustrating the timing of various signals that are generated at the time of execution of microprogram read-out and write insructions;

Figure 13 is a flow chart showing a typical program for introducing a signal delay by a software operation;

Figure 14 is a flow chart showing a modification of the program shown in Fig. 13;

Figure 15 is a block diagram showing essential portions of a modified echo chamber or machine; and

Figure 16 is a chart illustrating the movement of incremented read-out and write addresses in the memory.

Fig. 1 is a block diagram showing an embodiment of a basic system structure making use of a digital signal processor (DSP). In the present embodiment, at least certain portions of the DSP 1 and a memory control unit (MCU) 2, both of which are electronic elements or components, are designed as LSIs. A digital signal memory 3 has a storage capacity of (for example) 16 K (16,384) words or 64 K (65,536) words of digital signals, with each word comprising 24 bits. The digital signal memory 3 may for example comprise a D-RAM (dynamic-random access memory). A host computer system 4 includes a microprocessor designed for managing and controlling the digital signal processing to be executed by the DSP 1 and MCU 2. In the present embodiment, writing can be effected from the host computer system 4 into a microprogram memory 6 and a coefficient memory 7 through an interface circuit 5 in the DSP 1.

An address control unit 8 is provided in the MCU 2 for addressing respective words stored in the digital signal memory 3. The address control unit 8 includes an incrementer, comparator, etc. and is controlled by various control signals supplied from a sequence control unit 9 of the DSP 1. Other components of the MCU 2 include an interface circuit 10 for transmitting and receiving signals to and from the host computer system 4, and a scratchpad memory 11.

The digital signals to be processed by the system shown in Fig. 1 may for example be PCM audio signals or digital video signals etc. and comprise analog signals that are quantised so that each sampled value of the analog signal corresponds to one word of the digital signal, which word comprises for example 14 or 16 bits. It should be noted that the system making use of the DSP 1 is designed for processing 24-bit digital signals words in consideration of the increase in the number of bits and the resulting overflow to be caused by multiplication of the 14-bit or 16-bit digital signal words by coefficient data.

Fig. 2 is a block diagram showing the circuit structure of the DSP 1 in more detail. As shown in Figs. 1 and 2, the DSP 1 includes a 24-bit data bus DB that is connected to an arithmetic

processing unit 20, an input register 12, an output register 13 and a digital signal input/output (I/O) port 16. The input register 12 operates to convert serial data from a data input terminal 14 into parallel 24-bit data and to output the resulting data to the data bus DB. The output register 13 operates to convert parallel 24-bit data from the data bus DB into serial data and output the resulting data to a data output terminal 15. The arithmetic processing unit 20 includes at least an arithmetic logic unit (ALU) 21 and a multiplier 22, and a multiplexer 23 operatively connected to the ALU 21 and to the multiplier 22. A data memory 31 and a temporary register 32 are provided to act as a scratchpad memory in a data memory unit 30 in which intermediate data generated in the course of data processing in the arithmetic processing unit 20 are temporarily stored. The data memory 31 has a capacity of (for example) 256 words with each word comprising 24 bits. The coefficient memory 7, which is designed for storage of items of multiplier coefficient data each comprising one 12-bit word, has a capacity of 2 pages each comprising 512 16-bit words (16 bits × 1024 words). Access may be had to each word in the coefficient memory 7 by means of address information from a coefficient pointer 71. An output terminal of the coefficient memory 7 is connected to terminals of the multipliers 22 and 23, for supplying coefficient data X thereto, and is also connected to the 24-bit data bus DB, via a bi-directional buffer gate 41. The data bus DB is also connected through a bi-directional buffer gate 42 to a terminal of the multiplier 22 for supplying multiplicand data Y thereto. In addition, the data bus DB is connected to an output terminal of the data memory 31 and to an input terminal of the temporary register 32. In addition to the input terminal for the coefficient data X, the multiplexer 23 has an input terminal for output data TP from the temporary register 32, an input terminal for product data P from the multiplier 22, and an input terminal for data PP which is obtained by a logical 12-bit shift (or an arithmetical 11-bit shift) towards the right or lower side of the product data P. An output from the muliplexer 23 is transmitted to the ALU 21, which has bit shifting logic 24 operatively associated therewith. A status register 25 stores the contents of a flag that may be changed with the arithmetic processing carried out in the ALU 21. The 24-bit digital data resulting from the arithmetic operation in the ALU 21 is supplied to the data bus DB through a buffer gate 43 and to the data memory 31 through one input terminal of a multiplexer 33. The output data TP from the temporary register 32 is supplied to another input terminal of the multiplexer 33.

The microprogram memory 6 stores a microprogram for instructing a processing sequence to be carried out by the respective circuits in the DSP 1. A series of microinstructions that make up the microprogram are read out sequentially by address signals supplied from a sequencer 91. These microinstructions have a word length of (for example) 32 bits and are supplied through a pipeline register 61 to an instructions data bus IDB. The 32 bits of each microinstruction word are divided into several fields such as one for immediate data, one for controlling the data memory 31, one for controlling the ALU 21 and one for controlling the sequencer 91. The immediate data contained in the microinstructions are supplied through a buffer gate 44 to the data bus DB while control data for the data memory 31 is supplied through a multiplexer 34 to a lower address input port of the data memory 31. In the sequencer 91, the address in the microprogram memory 6 to be read out next time is determined by sequencer control data contained in the microinstructions and status data, such as flag status, from the status register 25. Output control logic 92 is designed for controlling the MCU 2 and is controlled by the microinstructions, the control logic 92 and the sequencer 91 making up the sequence control unit 9 (see also Fig. 1). The microinstructions also contain bits for incrementing a data pointer 35 designating the upper address of the data memory 30 and for incrementing the coefficient pointer 71.

Data B from the host computer system, in an eight bit per word format, can be written into the microprogram memory 6 and the coefficient memory 7 through the interface circuit 5.

Fig. 3 shows a memory map of the microprogram memory 6 and the coefficient memory 7 as viewed from the host computer system 4. As can be seen from Fig. 3, the memories 6, 7 may be viewed from the host computer as a memory MR with a total capacity of 4096 (2^{12}) words, with each word comprising eight bits (one byte). Access may be had to the memory MR byte per byte with a 12-bit address A. The memory MR is divide into two equal portions each comprising 2048 bytes. One of the portions with addresses $000 to $7FF in hexadecimal code is an area MPM for the microprogram memory 6, while the remaining portion with addresses from $800 to $FFF is an area CFM for the coefficient memory 7. The 2048 bytes of the coefficient memory area CFM are further divided into two equal portions. One such portion with the addresses $800 to $BFF is designated page 0 while the remaining portion with the addresses $C00 to $FFF is page 1. In this manner, access may be had from the host computer to the 4096 bytes having the addresses from $000 to $FFF, byte per byte, by addressing with the 12-bit address A. In the DSP 1, the program memory area MPM may be addressed by the sequencer 91 for simultaneously reading out 32-bit items of microinstruction data I, while the coefficient memory area CFM may be addressed by the coefficient pointer 71 for simultaneously reading out 16-bit items of coefficient data K. If the bits of the address A are designated as a_0 --

-- a_{11}, the nine bits a_2 ---- a_{10} are used as an address from the sequencer 91 for having access to the 512 32-bit words, whereas the nine bits a_1 ---- a_9 are used as an address from the coefficient pointer 71 for having access to the 512 words of page 1. One bit a_0 is used for page switching.

The 8-bit signals from the host computer system 4 are transmitted to the DSP 1 as four different types of signal by 2-bit mode-switching signals RS 0, RS 1. Fig. 4 shows the contents of the 8-bit signals for the four modes. The mode switching signals RS 0, RS 1 stand for four different states 00 to 11, namely a data mode M_0, an upper address mode M_1, a lower address mode M_2 and a control mode M_3. As may be seen from Fig. 4, in the data mode M_0 bits b_0 to b_7 of the 8-bit data B from the host computer system 4 respective bits d_0 to d_7 of 8-bit data D in practice written into the areas MPM and CFM. In the upper address mode M_1, the bits b_0 to b_3 are the upper four address bits a_8 to a_{11} of the 12-bit address for obtaining access to the memory MR. In the lower address mode M_2, the bits b_0 to b_7 are the address bits a_0 to a_7 of the lower 8-bit address. In the control mode M_3, the respective bits b_0 to b_7 of the upper 8-bit data B are used as respective control signals. For example, the bit b_7 is used as a page switching control signal PAGE for the coefficient memory area CFM.

Reference is now made to Figs. 5 and 6 for illustrating an operation of writing data into the memory MR from the host computer system 4.

Fig. 5 is a block circuit diagram of the interface circuit 5, program memory 6, coefficient memory 7 and neighbouring circuit elements in the DSP 1 shown in Figs. 1 and 2. The above-mentioned 8-bit data B, a chip select signal CS and the above-mentioned mode switching signals RS 0, RS 1 are supplied, respectively, to a data input port 17, a chip select input terminal 18 and mode switching signal input terminals 19 R_0, 19 R_1 showns in Fig. 5. A clock pulse signal CP as shown in Fig. 6 is supplied to a clock pulse terminal CLK shown in Fig. 5 for synchronising the operation of the circuit elements included in the DSP 1.

Suppose now that, in order to transfer data from the host computer system 4, the chip select signal $\overline{CS}$ is changed to a low level (L) and again changed to a high level (H) at a time t_1. A first register 51 connected to the data input port 17 is then enabled at the time t_1 to conduct the 8-bit data B from the data input port 17 and transmit the data as an output signal. The chip select signal $\overline{CS}$ is changed with a timing independent of that of the clock pulse signal CP. A $\bar{Q}$ output of a flip-flop 53, connected to a mode decoder 52 is turn connected to the input terminals 19 R_0, 19 R_1, is changed from "H" to "L" at a time t_2 corresponding to the first rise or positive-going transition of the clock pulse signal CP since the time t_1. This $\bar{Q}$ output is held at the low level "L" until a time t_3 corresponding to the next rise of the clock pulse signal CP, and is transferred to a terminal $\bar{G}_1$ of the mode decoder 52. The mode decoder 52 is responsive to the mode switching signals RS 0, RS 1 at the terminals 19 R_0, 19 R_1 for activating only one of the signal lines corresponding respectively to the modes M_0, M_1, M_2, M_3 and only during the time interval t_2 to t_3. Thus, for the data mode M_0, an eight-bit data latch circuit 54 is activated to latch the bits b_0 to b_7 and, for the control mode M_3, a control data latch circuit 56 is activated to latch only required ones of the bits b_0 to b_7. An address counter circuit 55 comprises three 4-bit presettable counters respectively corresponding to the address bits a_0 to a_3, a_4 to a_7 and a_8 to a_{11}. For the upper address mode M_1, the bits b_0 to b_3 are latched by one of the 4-bit presettable counters corresponding to the address bits a_8 to a_{11}. For the lower address mode M_2, the bits b_0 to b_3 and b_4 to b_7 are latched by two 4-bit presettable counters corresponding to the address bits a_0 to a_3 and a_4 to a_7, respectively. These latching operations are designed to occur at the time t_3.

As may be seen from the memory map shown in and described with reference to Fig. 3, the eight bits d_0 to d_7 of the one byte of data D from the data latch circuit 54 may be used from time to time as a byte i_0 to i_7, i_8 to i_{15}, i_{16} to i_{23} or i_{24} to i_{31} included in the microinstruction data I, or a byte k_0 to k_7 or k_8 to k_{15} of the coefficient data K. Which of these bytes corresponds to the eight bits d_0 to d_7 is determined by the address bits a_0, a_1 and a_{11} of the address A. Thus, whether the data is microinstruction data or coefficient data is determined by the address bit a_{11}. If the data is microinstruction data, the relevant one of the four bytes (32 bits) may be specified by the address bits a_0, a_1 and, if the data is coefficient data, the relevant one of the two bytes (16 bits) may be specified by the address bit a_0. Thus, the address bits a_0, a_1, a_{11} are supplied from the address counter 55 to an address decoder 57 from which six outputs corresponding to the bits i_0 to i_7, i_8 to i_{15}, i_{16} to i_{23}, i_{24} to i_{31}, k_0 to k_7 and k_8 to k_{15} are supplied to enable one of six 8-bit buffer gates connected to the output of the data latch circuit 54. Four of these six 8-bit buffer gates are provided in a byte selection circuit 62 for the microinstruction data, and the remaining two gates are provided in a byte selection circuit 72 for the coefficient data.

The microprogram memory 6 and the coefficient data memory 7 are configured with bytes as structural units. Thus, the microprogram memory 6 is formed by parallel connection of four 512-byte memory units and therefore has a capacity of 512 words each comprising 32 bits. The coefficient memory 7 is formed by parallel connection of two 1024-byte memory units and thus has a capacity of 512 words in 2 pages each comprising 16 bits. The outputs from the

four 8-bit buffer gates of the byte selection circuit 62 are supplied to the four 512-byte memory units of the microprogram memory 6, while the outputs from the two 8-bit buffer gates of the byte selection circuit 72 are supplied to the two 1024-byte units of the coefficient data memory 7. Referring to the 12-bit address output A from the address counter 55, the nine bits a_2 to a_{10} are supplied to an address bus of the microprogram memory 6 through a nine-bit buffer gate 63, while the ten bits a_1 to a_{10} are supplied to a coefficient address bus of the coefficient memory 7 through a ten-bit buffer gate 73.

The buffer gates 63, 73 are tristate buffers, that is they are enabled by a refresh signal REFRESH switched upon execution of refresh instructions of the microprogram and otherwise remain in a high impedance or floating state. The address decoder 57 also operates in response to the refresh signal REFRESH for activating one of the six decoder outputs so as to enable only the relevant one of the 8-bit buffer gates of the byte selection circuits 62, 72. The buffer gates of the byte selection circuits 62, 72 are also designed as tristate buffers.

The 9-bit program memory address relevant to the address bits a_2 to a_{10} is then outputted from the sequencer 91 for sequentially addressing the words in the microprogram memory 6 in order to read out the microinstructions. Fig. 6 shows microinstructions MPI read out sequentially from the microprogram memory 6. Numbers ---, N − 1, N, N + 1, --- are affixed to a series of instructions used for controlling the MCU 2 and the respective circuits included in the DSP 1. It should be noted that the N'th instruction includes a 32-bit refresh instruction in which a refresh command bit is active, and that the instruction immediately following the N'th instruction is disregarded in the DSP 1 because of the resulting refresh operation. Therefore, a no-op instruction (NOP instruction) is inserted between the N'th and (N + 1)'th instructions that are intended for actually controlling the signal processing operation.

Referring still to Fig. 6, the N'th instruction including the refresh instruction is read out from the microprogram memory 6 at a time t_{11} determined by the clock pulses CP. This N'th instruction is delayed by one clock pulse interval by being passed through the pipeline register 61, and is executed within the next clock pulse interval t_{12} to t_{13}. During this time interval t_{12} to t_{13}, the microprogram memory 6 and the coefficient memory 7 are in the memory access or addressing mode such that access or addressing from the sequencer 91 and the coefficient pointer 71 resulting from the internal operation of the DSP 1 is inhibited and the 8-bit data from the data latch circuit 54 can be written in the 8-bit word addressed by the address A from the address counter 55. In addition, during the time interval t_{12} to t_{13}, the status is such that latching by the pipeline register 61 of the 32-bit data read out from the microprogram memory 6 is inhibited, a pipeline register enable signal ($\overline{\text{ENABLE}}$) being at a high level. Therefore, the NOP instruction that is included in the instruction MPI read out from the microprogram memory 6 (that is, the instruction directly following the refresh instruction) is not latched by the pipeline register 61, and the output from the pipeline register 61 is such that the N'th instruction persists during the time interval t_{12} to t_{14}. In this manner, a refresh operation is performed during the time interval t_{12} to t_{13}, the refresh signal REFRESH being active, so that the buffer gates 63, 73 are enabled and access is obtained to one word stored in the microprogram memory 6 or the coefficient memory 7. One of the buffer gates in the byte select circuits 62, 72 is enabled for writing a relevant one-byte item of data from among the 4096 bytes stored in the memory MR.

The coefficient pointer 71 for addressing the coefficient memory 7 in response to execution of the microprogram in the DSP 1 is designed for sending a 9-bit address output corresponding to the address bits a_1 to a_9 to a coefficient address bus in order to address one of the two 512-word pages of the coefficient memory 7, and the signal corresponding to the address bit a_{10} specifying the page 0 or the page 1 is outputted in response to the bit b_7 in the 8-bit item of data supplied from the host computer system 4 during the control mode M_3. Thus, during the control mode M_3, the 8-bit item of data is latched by the control data latch circuit 56, while the page switching signal PAGE outputted in response to the bit b_7 is supplied to a data input terminal D of a D-type flip-flop 74. A refresh signal REFRESH inverted during execution of a specified instruction in the microprogram, such as a refresh instruction, is supplied to a trigger input terminal T of the D-type flip-flop 74, the data at the data input terminal D being introduced at the refresh timing and outputted from the Q output terminal. This output of the Q output terminal of the flip-flop 74 is supplied as the address bit a_{10} to the coefficient address bus through a buffer gate 75. Therefore, page switching is performed only upon execution of a specified instruction such as a refresh instruction contained in the microprogram and in dependence upon the contents of control data originating from the host computer system 4.

In the above-described digital signal processing system, the microprogram memory 6 and the coefficient memory 7 appear as a continuous memory MR when viewed from the host computer system 4, as shown in Fig. 3, and may be addressed with a predetermined series of address bits a_0 to a_{11}. In this manner, data transfer may be performed easily and positively. In addition, the coefficient memory 7 has at least two pages corresponding to the total memory area addressable during execution of the microprogram execution at the DSP 1, and the page switching is

controlled by control data from the host computer system 4 (for example, the contents of the data bit b_7 for the above-mentioned control mode M_3). In this manner, a portion of a series of coefficients need not be rewritten, as in conventional practice, so that adverse effects caused by noise or oscillation may be prevented from occurring. Moreover, since the timings for writing data from the host computer system 4 into the memory MR and page switching are provided in a specified cycle that is irrelevant to (i.e. independent of) the digital signal processing such as logic operation or multiplication carried out in the course of the microprogram execution, as for example in a refresh cycle accompanying refresh instruction execution, thereis no risk of, for example, coefficients being changed in the course of multiplication.

Reference is now made to Fig. 7 for illustrating multiplication of double precision coefficient data to be effected at the multiplier 22.

A 12-bit item of coefficient data X is supplied to a coefficient input terminal X of the multiplier 22, as mentioned above, and is multiplied by a 24-bit item of digital signal data Y supplied to a multiplicand input terminal Y to give a 36-bit product. The upper 24 bits are taken out as product data P and supplied to the multiplexer 23. When 24-bit double precision is required of the coefficient data, these 24 bits are divided into an upper 12-bit item of data (one word long data) X_H and a lower 12-bit item of data (one word long data) X_L. The items of data X_H and X_L are respectively multiplied by the multiplicand data Y and the respective products are added together to give a product for 24-bit coefficient data. In this case, since the upper 12 bits of the 36-bit item of product data $X_L.Y$, obtained upon multiplication of the lower 12-bit item of data X_L by the multiplicand data Y, corresponds to the lower 12 bits of the 24-bit item of product data P, it is necessary to use a second item of product data PP in which the product $X_L.Y$ is shifted one word or 12 bits towards the lower side and the upper 12 bits are expanded 12 bits towards the upper side to give a 24-bit item of data.

In more detail, referring to Fig. 7, when the 24-bit digital multiplicand signal Y is multiplied by the 24-bit double precision coefficient data, an item of coefficient data X_H corresponding to the upper 12 bits of the 24-bit coefficient data is multiplied by the multiplicand data Y in the first stage operation to give a 36-bit product $Y.X_H$, and the upper 24 bits of the product $Y.X_H$ are taken out from multiplier 22 as first product data P. As a second stage operation, the coefficient data X_L corresponding to the lower 12 bits of the aforementioned 24-bit coefficient data is multiplied by the data Y to give a 36-bit product which is then shifted by 12 bits to the lower side so that the product $Y.X_L$ is now placed as a string shown by a double-dotted chain line in Fig. 7. The upper 12 bits of the product are subjected to sign expansion and the resulting 24 bits are taken out as product data PP from the multiplier 22. "Sign expansion" means an operation in which the sign bits of the digital data represented in 2's complement format are added in the upper side by a number equal to the total bit number of the new data string less the bit number of the original product.

Table 1

decimal number	Binary number in 2's complement	
	4-bit format	8-bit format
7	0 1 1 1	0 0 0 0 0 1 1 1
6	0 1 1 0	0 0 0 0 0 1 1 0
5	0 1 0 1	0 0 0 0 0 1 0 1
4	0 1 0 0	0 0 0 0 0 1 0 0
3	0 0 1 1	0 0 0 0 0 0 1 1
2	0 0 1 0	0 0 0 0 0 0 1 0
1	0 0 0 1	0 0 0 0 0 0 0 1
0	0 0 0 0	0 0 0 0 0 0 0 0
-1	1 1 1 1	1 1 1 1 1 1 1 1
-2	1 1 1 0	1 1 1 1 1 1 1 0
-3	1 1 0 1	1 1 1 1 1 1 0 1
-4	1 1 0 0	1 1 1 1 1 1 0 0
-5	1 0 1 1	1 1 1 1 1 0 1 1
-6	1 0 1 0	1 1 1 1 1 0 1 0
-7	1 0 0 1	1 1 1 1 1 0 0 1
-8	1 0 0 0	1 1 1 1 1 0 0 0

In the example shown in Table 1 above, 4 bits (for example) equal to the uppermost bit (sign bit) 0 or 1 of the 4-bit binary data in 2's complement representation are added to the upper side to give an 8-bit item of data which stands for the same number. The hardware can be so designed that the data signal line between the multiplier 22 and the multiplexer 23 is connected as shown in Fig. 2 whereby the first product data P and the second product data PP may be obtained without any changes in the multiplying operation at the multiplier 22. The multiplexer 23 sends the data P to the ALU 21 at the time of first stage multiplication and the data PP to the ALU 21 at the time of second stage multiplication so that the data P and PP are added together in the ALU 21. At least the upper 12 bits of the product may be transferred to the PP input terminal of the multiplexer and the bit shift as well as sign expansion may be performed in the ALU 21.

It may be seen from the foregoing that the multiplier 22 need only have a circuit scale or functional capacity of 24 × 12 bits for a digital signal data length of 24 bits and a coefficient data word length of 12 bits and that, for an algorithm necessitating a double-precision coefficient word length of 24 bits, the double precision coefficient word is divided into an upper 12 bits and a lower 12 bits that are multiplied respectively by the digital signal data, with the product resulting from the second multiplication being shifted by one word or 12 bits by logic shift or 11 bits by arithmetical shift to the lower side to give second product data PP which is then added to the product P of the first multiplication by way of completing the multiplication of

24 bits by 24 bits. The increase in the number of cycles to be executed is approximately only one, thus providing extremely fast operation as compared to a conventional double precision operation.

In addition, single-accuracy fast operation may be selected for operating components requiring only 12 bit coefficient accuracy, and the double-precision may be selected for operating components requiring higher coefficient precision, such as a digital filter. The result is a signal processing system with improved hardware exploitation.

The present invention is not of course limited to the above-described embodiment. For example, the word lengths of the digital signal or coefficient data can be preset to any desired value. The timing for the data writing or page switching operation need not be provided in the refresh cycle, but may instead be provided in the cycle of execution of instructions such no-op or transient stop instructions (pose instruction) that are to some extend irrelevant to (independent of) the actual processing of the digital signals. Many other changes may be made within the scope of the invention.

A digital signal delay circuit which makes use of the digital signal processing system to provide an echo chamber or machine, and especially the structure of the memory control unit, will now be described with reference to Figs. 8 to 14 of the accompanying drawings.

The numeral 3 in this arrangement designates a signal delay memory (SDM) having a capacity of (for example) 64 K (65,536) 24-bit digital words or signals. Each word stored in the SDM 3 is addressed by a 16-bit (for example) memory address MA from an address management unit 8 (hereafter abbreviated to AMU) included in the memory control unit 2.

As shown for example in Fig. 9, the 64 K words stored in the SDM 3 are divided into n memory cells C_1 to C_n. An address management memory (AMM) 81 is included in the AMU 8 for storage of boundary addreses (top address TA and bottom address BA) of the first to n'th memory cells C_1 to C_n and the addresses of the cells being accessed, or current addresses, CA. The cells C_1 to C_n in the SDM 3 may have overlapping words, as do the cells C_2 and C_3, or may have interrupted address area, as do the first cell C_1 and the second cell C_2. For each of the memory cells C_1 to C_n, the AMM 81 has an area 81B for storage of the bottom address BA, which is the minimum address value, an area 81T for storage of the top address TA, which is the maximum address value, and an area 81C for storage of current addresses CA in operation and ranging from the address BA to the address TA. Each area 81B, 81T, 81C may store words in numbers related to the numbers of the cells and may be addressed by the serial cell numbers.

Writing of the addresses BA, TA and CA (initialisation or subsequent changes) may be performed by the host computer system 4 as in the case of the DSP 1. The computer system 4 outputs (for example) 6-bit data for specifying the serial numbers of the memory cells and 16-bit address data for identifying the addresses BA, TA and CA. The serial cell number data and the address data are supplied to the AMU 8 through a multiplexer 82 used as switching and selecting means in the MCU 2 and through an address register 11, respectively. It should be noted that, for 6-bit serial cell number data, the SDM 3 can be divided into up to 64 memory cells.

Fig. 11 shows a typical circuit for the AMU 8. In Fig. 11, an adder 83 is designed to add "1" to (i.e. increment) the current address CA read out from the area 81C of the AMM 81. The result of addition from the adder 83, that is, the incremented current address data, is supplied to a comparator 84 and a multiplexer 85. The result of addition and the top address TA read out from the area 81T of the AMM 81 are compared in the comparator 84 and the result of the comparison is supplied to a switching control terminal of the multiplexer 85. The multiplexer 85 is operative to select and output the bottom address BA from the area 81B of the AMM 81 or the result of addition from the adder 83 in view of the above-mentioned result of comparison. Thus, the bottom address BA is selected and outputted when the result of addition is larger than the top address TA. The output address data from the multiplexer 85 is supplied through a multiplexer 86 to the area 81C of the AMM 81 and written in response to a write instruction WT from the microprogram memory 6. In this manner, the current address CA in the memory area 81C is incremented each time the write instruction WT is outputted to the SDM 3 from the microprogram, and the address CA is again incremented from the bottom address BA upon reaching the top address.

It should be noted that the DSP 1 shown in Fig. 1 corresponds to the circuit portions shown in Fig. 8, excluding the MCU 2, SDM 3 and the host computer system 4.

In the present embodiment, the microinstructions include a field for controlling the AMU 8 intended for management of the SDM 3. With up to 64 memory cells of the SDM 3, 6 bits are required for specifying the serial cell numbers and 2 bits are also required for controlling the read and write operations of the SDM 3. Thus, the field is an 8-bit AMU or SDM control field. This 8-bit AMU control field signal is supplied to the AMU 8 where the 16-bit current address CA is produced for addressing the SDM 3 from the current address area 81C of the AMM 81.

When (for example) the first memory cell C_1 in the SDM 3 is used as a delay circuit, it is necessary to write in advance the bottom address BA_1 and top address TA_1 for the memory cell

C_1, as well as the current address CA_1 ranging from BA_1 to TA_1, in the respective words stored in the areas 81B, 81T and 81C in the AMM 81 and having the addresses for the serial cell numbers equal for example to "1" (an initialising step). During such initialising, the multiplexer 82 in the MCU 2 is switched to the host computer system 4 which then transmits a signal specifying the serial cell number "1" and address data for the addresses BA_1, TA_1 and CA_1 sequentially to the AMM 8. When transmitting the addresses BA_1, TA_1 and CA_1 sequentially, the computer system 4 also transmits the address identification codes that are used for identifying these addresses. The data thus transmitted is written into the areas 81B, 81T and 81C.

After thus initialising the memory cells to be used, the multiplexer 82 is switched to the microprogram memory 6 so that the AMU 2 is controlled by the microprogram. In this case, control proceeds to a digital signal delay loop, after all the words in the SDM 3 are cleared or set to "0". In this delay loop, read and write operations are performed for the words in the SDM 3 addressed by the current address CA, while the address CA is incremented. In the microprogram, it is only necessary to designate the serial number of the memory cells and to issue read and write instructions, the operations such as incrementing the current address CA and switching to the bottom address BA after reaching the top address being performed automatically in the AMU 2.

When the serial cell number in the SDM 3 is specified in the microprogram digital signal delay loop, the current address CA of the word corresponding to the cell number of the AMM 81 is read out and access is made to the SDM 3 by this address CA. When the SDM 3 has an access time of the order of or less than one microprogram instruction cycle, the data contents of the addressed word can be inputted or outputted by the following microinstruction whereby digital signals can be read out from or written into the SDM 3. In the read mode, the current address CA is not updated in the AMU 2 and the data read out from the SDM 3 in the instruction cycle subsequent to designation of the serial cell number may be transmitted through the data bus DB to the registers or to circuit elements performing the next processing steps, such as a multiplier and a D/A converter. In the write mode, a write pulse is outputted in response to the microinstruction subsequent to the memory access whereby the digital data on the data bus DB is written into the addressed word stored in the SDM 3, while the current address CA is updated in the AMU 8, that is, the address data from the multiplexer 85 is taken into the AMU 8.

Reference is now made to Fig. 12 for illustrating the operational timing in the write mode. In Fig. 12, a time interval T_S corresponds to one instruction cycle in the microprogram. The serial cell number is designated in relation to the write instruction during a time interval t_1 to t_2, and data is exchanged between the SDM 3 and the data bus DB during a time interval t_2 to t_3. When the serial cell number is specified at a time t_1, the current address CA read out from the current address area 81C of the AMM 81 is determined at a time t_{11} after the lapse of a predetermined access time. At a time t_{12}, right after the time t_{11}, an address strobe pulse for the SDM 3 is produced to permit access to the SDM 3. Reading from and writing into the SDM 3 is enabled after lapse of an address time which is determined by the characteristics of the memory device being employed and is of the order of one hundred and several scores of nanoseconds with a dynamic RAM. Write and read pulses are outputted at a time t_{13} directly before the lapse of a time t_2 to t_3 for the next instruction cycle or directly before a time t_3 for the exchange of digital signal data between the addressed word and the data bus DB. During the time interval from the time t_{11} at which the current address CA is determined until the time t_{13} at which the above write pulse is issued from the SDM 3, incrementing the address CA by the adder 83, comparison thereof with the top address TA by the comparator 84 and selection by the multiplexer 85 are performed in the AMU 3. The address from the multiplexer 85, that is the next address NA necessary for obtaining access to the SDM 3 in the next signal delay loop, is sent through the multiplexer 86 to the current address area 81C. The next address NA is written into the current address area 81C only at the time t_{13} at which the write pulse is outputted in the write mode. Therefore, in one cycle of the microprogram signal delay loop, the digital signal is written into the same word in the SDM 3 to which access has been obtained during reading with the current address CA, and the current address CA in the current address area 81C of the AMM 81 is written into the next address only at this time. It is after the total words in the memory cell used as a delay line (such as the first memory cell C_1) have been addressed that the word into which the digital signal has been written is read out a second time, the delay time being a product of the total number of words included in the memory cell or the difference between the top address TA and the bottom address BA and the cycle time for the microprogram signal delay loop. The cycle time of the loop may be equated to a sampling period when a program is inserted into the signal delay loop for repeatedly checking for termination of a sampling operation at an A/D converter 101 until termination of the sampling.

For a sampling clock frequency of 50 kHz (a sampling period of 20 microseconds), and if the number of words in the memory cell used as a delay line is equal to 1,000, the delay time is 20 milliseconds. For an increase of one word in the memory cell, the delay time is increased by 20

microseconds and, for a decrease of one word in the memory cell, the delay time is decreased by 20 microseconds. The number of words can be easily increased and decreased by rewriting at least one of the bottom address BA and the top address TA of the memory cell in an operation controlled from the host computer 4.

Fig. 13 shows a typical flow chart for realising such signal delay procedure by a microprogram. Before proceeding to the signal delay loop, "0" is written into all of the words in the SDM 3 in a step 201 (all clear or initial resetting). This step is followed by the signal delay loop beginning from a step 202 which makes a check that the A/D conversion is completed. Each time the sampling operation is performed in the A/D converter 101 the digital signals are read from or written into the SDM 3 while the current address CA in the current address area 81C of the AMM 81 is automatically rewritten into the AMU 8 (step 203 *et seq*). In a step 203, the digital signal data that has been subjected to A/D conversion is stored through the data bus DB in (for example) a register area Ra of the data memory unit 30. In steps 204 and 205, serial memory cell numbers are allotted to the SDM 3 for reading out the digital signal data. Thus, in the step 204, the instruction to read the first memory cell C_1 is issued to the AMU 8 which then outputs a current address CA corresponding to the first memory cell, such address CA being used for obtaining access to the SDM 3. Output data may be valid and read out from the SDM 3 after an access time determined by the characteristics of the memory elements used in the SDM 3. The output data may become valid in general in one hundred and several scores of nanoseconds in the case of using a dynamic RAM as the SDM 1. In the step 205, with the output data fixed in this manner, the data supplied from the SDM 3 to the data bus DB is stored in (for example) a register area Rb of the data memory unit 30. Therefore, reading from the SDM 3 is realised in two microinstruction steps. However, the reading from the SDM 3 may also be realised by one microinstruction step when the time for executing one microinstruction step (instruction cycle) is long or the access time to the AMM 81 or the SDM 3 may be minimised.

In a step 206, which is a write instruction for the first cell in the SDM 3, a write instruction and serial cell number designation are issued to the AMU 8, whereby the current address CA corresponding to the first cell is read out for addressing the SDM 3. The write signal for the SDM 3 is outputted in the next step (207). At this time, the contents of the register area Ra storing the data from the A/D converter 101 are supplied to the data bus DB and written into the SDM 3.

The data read out from memory cell C_1 of the SMD 3 and stored in the register area Rb is supplied to a D/A converter 100 in a step 208. Then, the control returns to the step 202 to complete the basic signal delay loop.

The above-mentioned echo chamber or machine can be constructed from several signal delay lines without changes in hardware design by introducing into a step 209 (or in a position shown by dotted lines in Fig. 3) a program in which other memory areas are designed for read/write operations or a program in which data read out from respective cells of the SDM 3 are multiplied by a coefficient and the resulting product is added to the data before delay (such as data subjected to A/D conversion and stored in the register area Ra). In addition, the serial memory cell number may be designated by the host computer system 4 during, for instance, the waiting time for A/D conversion in the step 201 for rewriting the above-mentioned bottom address BA or top address TA for dynamically changing the delay time of the signal delay circuits corresponding to the respective cells.

The data from the A/D converter 101 can be written directly into the SDM 3 or the data read out from the SDM 3 can be forwarded directly to the D/A converter 100 without making use of the register areas Ra, Rb of the data memory unit 30. Fig. 14 shows a flow chart for this procedure. In Fig. 14, the steps 201, 202 are the same as in Fig. 13. However, a step 301 for reading (for example) the first memory cell C_1 of the SDM 3 (corresponding to the step 204 in Fig. 13) is carried out next to the step 202. When the data thus read out is valid in the data bus DB, it is forwarded directly to the D/A converter 100 in a step 302. In a step 303, which is equivalent to the step 206 in Fig. 13, access is made to the write address in the first memory cell C_1 of the SDM 3. In a step 304, the data from the A/D converter 101 is written via the data bus DB into the address to which access was obtained in the preceding step. The steps from the step 202 for judging the condition through to the step 304 make up a complete cycle of the delay loop. It should be noted that the condition step 202 may be placed between the steps 302 and 303. In this case, the steps 301, 302, 202, 303 and 304 make up one complete cycle of the delay loop.

A modified form of the above digital signal processor, when applied to an echo chamber or machine, will now be described with reference to Fig. 15.

In the present embodiment, a write address WA and read address RA shown in Fig. 10 are used instead of the current address CA. The addresses WA, RA are stored in areas 81W, 81R in the AMM 81, respectively.

The AMU 8 includes a WD memory circuit 8W for storage of write data WD to be supplied to

the write address WA, an RD memory circuit 8R for storage of read data D to be supplied to the read address RA, and an adder 83′ for adding the write address WA from the area 81W of the AMM 81 and the write data WD from the memory circuit 8W to one another during writing and adding the read address RA from the area 81R of the AMM 81 and the read data RD from the memory circuit 8R to one another during reading. The sum from the adder 83′ is supplied to a comparator 84 and to a multiplexer 85 used as switching and selecting means. The sum and the top address TA from the area 81T are compared with one another in the comparator 84 and the result of such comparison is sent to a switching control terminal of the multiplexer 85. The multiplexer 85 selects and outputs the bottom address BA from the area 81B of the AMM 81 or the sum from the adder 83′ depending on the result of the comparison. Thus, it selects and outputs the bottom address BA when the sum is larger than the top address TA, and the sum if the sum is not larger than the top address. The output data from the multiplexer 85 is the next address NA specifying the write or read address to which access is to be obtained next time. This output data is sent to the area 81W of the AMM 81 through a multiplexer 86W during writing and to the area 81R of the AMM 81 through a multiplexer 86R during reading. Write data "1" or write data from the host computer system 4 such as "0" or "2" have been supplied by this time to the memory circuit 8W for the write data WD to the write address WA, after switching and selection by a multiplexer 86′W, whereas read data "1" or read data from the host computer system 4 have been supplied similarly to the memory circuit 8R for the read data to the read address RA after switching and selection by a multiplexer 86′R. When the delay time is constant, the write data "1" and read data "1" are supplied from the multiplexers 86′W, 86′R to the memory circuits 8W, 8R so that the write and read data WD, RD from the memory circuits 8W, 8R are "1". Therefore, in the adder 83′, the write address WA is increased by one (incremented) for each writing operation, and the read address RA is also increased by one for each reading operation.

Reference is again made to Fig. 12 for illustrating the timing relationship of the reading and writing of the digital signals relative to the SPM 3. During a reading operation, the serial cell number is specified at the time t_1 and the read instruction is given to the AMU 8. The word in the read address area 81R of the AMM 81 corresponding to the specified serial cell number is addressed and the output data (read address RA) is valid at the time t_{11} so that the SDM 3 is addressed with the read address RA.

In the instruction cycle comprising the time t_2 to t_3 in Fig. 12, a read pulse is outputted at the time t_{13} when the access time had elapsed and the output data has become valid, thereby opening an output gate of the SDM 3 and supplying the contents of the addressed word on the data bus DB for further processing in registers or other relevant circuits such as a multiplier, adder or D/A converter.

During such reading operation, the read address RA is updated automatically in the AMU 8. When fixed at the time t_{11}, the read address RA from the area 81R is sent to the adder 83′ where it is added to the read data from the memory circuit 8R. For a constant delay time, the read data is "1" and the read address RA is incremented and supplied to the comparator 84 and multiplexer 85. When the read address thus incremented is less than the top address TA, it is selected at the multiplexer 85 as the next address NA and forwarded through the multiplexer 86R to the area 81R of the AMM 81. The above operation takes place automatically until the time t_{13}. The next address data is then taken into the area 81R in response to the read pulse produced at the time t_{13} for rewriting or updating the read address RA. When the incremented address from the adder 83′ is larger than the top address TA, the bottom address BA is selected by the multiplexer 85 as next address NA.

The operation during the writing phase of the microprogram is the same as the read operation described above if the read address RA, area 81R, memory circuit 8R, read data RD and the multiplexer 86R referred to in the description of the read operation are replaced by the address WA, area 81W, memory circuit 8W, write data WD and the multiplexer 86W, respectively. Thus, in response to the write pulse at the time t_{13}, the write address WA is updated while the digital signal data from the registers and data output circuits etc. is written into the SDM 3 via the data bus DB.

Fig. 16 illustrates the movement of the read address RA and write address WA in the memory cell of the SDM 3 used as the delay circuit. Supposing that the read address RA and write address WA shown in Fig. 16A are moved for each increment in the direction of the arrows, i.e. in the direction from the bottom address BA towards the top address TA, the product of the sampling period with the number of words N equal to the difference between the addresses RA, WA ($N = WA - RA$) represents the delay time. Fig. 16B shows the state of the memory cell in which the write address WA has once reached the top address where it is switched to the bottom address BA from which it is incremented again. The delay time is not changed since the sum of the number of words from the read address RA to the top address TA and the number of words from the bottom address BA to the write address WA is equal to the word number difference N.

In the foregoing description, it has been assumed that the delay time is constant and is not changed. A description will now be given of how the delay time can be changed during microprogram execution.

When it is desired to prolong the delay time, write data "0" is written into the memory circuit 8R from the host computer system 4 through the multiplexer 86'R at a time in the course of the signal delay loop well before addressing the SDM 3, for example during the waiting time for A/D conversion (the step 202 in Fig. 13). In this case, since "0" is added to the read address RA during the reading phase in the same cycle of the program loop (steps 204, 205), the read addres is not incremented and is inhibited from moving in the direction of the arrow in Fig. 16. On the other hand, the write address WA is incremented during the writing phase in the same cycle of the program loop, resulting in an incremented word number difference and the delay time being prolonged by one sampling period. From the next cycle on, both the read address RA and the write address WA are incremented so that a word number difference equal to (N + 1) is maintained. When it is desired to prolong the delay by n sampling periods, the aforementioned operation of writing "0" into the memory circuit 8R is repeated n times at intervals of the aforementioned sampling period (one cycle time of the program loop). The delay time can also be prolonged by writing "2" or higher values as write data into the memory circuit 8W. However, this is not preferred because the write address WA of the SDM 3 is increased by two or more at a time, resulting in the contents of intermediate words remaining unchanged and completely discrete read-out data.

When it is desired to shorten the delay time, write data "0" is written into the memory circuit 8W from the host computer system 4 through the multiplexer 86'W for reducing the delay time by one sampling period. This operation can naturally be repeated n times for shortening the delay time by n sampling periods.

In this manner, it is possible to change the initially set delay time by time intervals equal to an integral or whole number multiple of the sampling interval by writing "0" into the memory circuits 8R or 8W a desired number of times. In the above procedure, the read address RA or the write address WA is inhibited only temporarily from being incremented so that temporal continuity of the readout data may be maintained and noises may be prevented from occurring.

It can be seen from the foregoing that plural digital signal delay lines can be realised with the use of a single SDM 3 and the hardware design thus may be simplified in that the separate hardware devices needed for respective delay lines as required when using shift registers may be dispensed with. In addition, actual address management for the SDM 3 is taken charge of in the AMU 8 and the read and write addresses (current address) can be incremented or otherwise processed in the AMU 8 so that the ALU 21 (for example) may take over and execute other jobs and the throughput of the digital signal processing unit as a whole may be improved. The delay time for the signal delay line for each memory cell can be changed easily by software control through the host computer system 4, while the delay time can also be changed dynamically in the course of the real time signal processing.

Reference should be made to our copending application no. 8302850 (Publication No. GB 2 115 588 A) from which the present application was divided out.

CLAIMS

1. A digital signal processing system comprising a digital signal processing unit including at least a microprogram memory for storing a series of microinstructions for instructing a digital signal processing procedure, and a coefficient memory for storing coefficient data required for performing a series of arithmetical operation on the digital signal data, a memory control unit, at least one memory block for storage of digital signals, and a host computer system, wherein
data may be transferred and written into the microprogram memory and the coefficient memory from the host computer system,
the coefficient memory has at least two pages corresponding to the total memory area to which access may be had during digital signal processing to be effected in response to the microinstructions,
page selection of the coefficient memory may be performed under control from the host computer system,
the memory control unit has an address management memory for dividing said one memory block into a plurality of memory cells identified by respective addresses,
means are provided for writing boundary addresses of the memory cells from the host computer system into the address management memory in advance, and
memory access is made with serial memory cell numbers designated by the digital signal processing unit.

CLAIMS

Amendments to the claims have been filed, and have the following effect:–

Claim 1 above has been deleted.

New or textually amended claims have been filed as follows:–

1. A digital signal processing system comprising a digital signal processing unit including microprogram memory means for storing a plurality of microinstructions for instructing a digital signal processing procedure,
coefficient memory means for storing coefficient data required for performing a series of arithmetical operations on a digital signal,
memory control means,
at least one memory block for storing said digital signal,
a host computer system, and
means for writing data into the microprogram memory means and the coefficient memory means from the host computer system,
wherein the memory control means includes address management memory means for dividing said at least one memory block into a plurality of memory cells identified by respective boundary addresses,
means are provided for writing the boundary addresses of the memory cells from the host computer system into the address management memory means, and
the memory cells are accessible in accordance with serial memory cell numbers designated by the digital signal processing unit.

Printed in the United Kingdom for Her Majesty's Stationery Office, Dd 8818935, 1985, 4235.
Published at The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.